L2: Entry 1 of 2 File: USPT Sep 28, 1999

DOCUMENT-IDENTIFIER: US 5960200 A
TITLE: System to transition an enterprise to a distributed infrastructure

Detailed Description Text (83):
The database server communicates with the database through ANSI SQL queries. The database server implements each type of IMS DL/1 function as follows. The database server first validates the arguments to the IMS DL/1 function. It then dynamically creates an ANSI SQL query string. In the preferred embodiment of the present invention, this query string is forwarded to the database using the Oracle Call Interface (OCI) from Oracle Corporation. At a high-level, the process consists in initializing bind and define variables, setting currency information, executing the SQL query, and returning query status.

Detailed Description Text (103):
For more complex distributed applications, a preferred embodiment of the present invention can use tools based on DCE, which is a more comprehensive distributed system infrastructure that includes directory services, distributed security, multi-threading, distributed file system, and central time management, in addition to RPCs. As sample implementation of RPCs effective over both TCP/IP and DCE networks, is the Entera toolkit from Open Environment Corporation.

Detailed Description Text (107):
Target user interface definitions 213 can take one of three forms: database files 246, a header file 247, and a GUI file 248. Database files 246 contain the set of statements necessary to populate user interface repository 152 with screen and message information for MFS file 211. In a preferred embodiment of the present invention, the database files 246 can be viewed as ANSI SQL scripts.

Detailed Description Text (116):
The code generator 245 can be viewed as a large switch that, given a statement type, calls the appropriate function to generate "target" code for this statement. The code generator 245 relies on a library of functions that handle the actual code generation for the entire set of statements of the source language. One such library exists for each source language, with ANSI SQL, ANSI C, and Microsoft Visual Basic as the target languages.

Detailed Description Text (127):
In addition to procedural constructs and data elements and structures, a subset of the rules specializes in the extraction and transformation of external data access commands into application-independent external data access commands. In a preferred embodiment of the present invention, application-independent external data access commands could be encoded using the ANSI-SQL language. External data access operations are thus parsed and transformed into separate, application-independent data access commands. These generated data access commands are stored in at least one separate output file.

Detailed Description Text (141):
In a preferred embodiment of the present invention, the target DDL 238 is a relational database schema specified using conventional ANSI SQL language. Such a schema defines the tables that compose an application, along with their key fields, and other descriptive fields. Initial values and other constraints such as referential integrity clauses may also be included in this schema. Because relational schemas are well understood, and ANSI SQL syntax is well-documented, the primary task of the DDL converter 235 is to map the syntax of source DDL 232 to the corresponding ANSI SQL syntax. In a preferred embodiment of the present invention, this

"target" DDL 238 can be viewed as an intermediary language that can then be converted to the final target DDL language for increased maintainability and flexibility, as was the case with the user interface and procedural language conversion utility. For illustrative purposes, IMS DL/1 can be considered as the source DLL 232.

Detailed Description Text (151):
FIG. 27 is a block diagram of the data converter 236 of FIG. 24. A DBDxx.dlp file 231a is provided as input to the data converter 236, along with a flat file 231b that contains the data to be converted. The data converter 236 uses the data loading pattern specified in the DBDXX.dlp file 231a to determine the desired target format for each data record in the data flat file 231b and generate an ANSI SQL file 239a containing the INSERT statements necessary to populate the target database with the data provided.

Previous Doc Next Doc Go to Doc#